

time on the system, the subsystem and function that are presently causing an output to be executed to the trace data file, and the trace level that corresponds to the calling trace code as indicated by step 601 in FIG. 6. The message text present in the calling trace statement is then formatted as indicated by step 602 in FIG. 6. This formatting of the message text usually includes the replacement of token words that are present in the calling code with definitions that correspond to the particular situation encountered by this code. For example, in the above illustrated trace statement relating to a particular department number, the specific department number that was encountered by the code is included within the message text.

The code of the trace library then makes a determination as indicated by decision step 603 in FIG. 6 as to whether or not a file is open to receive the trace data. If it is, the trace message is immediately coupled to that file by way of a code that is represented by step 606. If the file is not open, however, a file is first opened as indicated by step 604 and the file-open flag 114 is set to its true condition as indicated by step 605 prior to outputting a trace message. After the message is delivered to the data file there is a return from the trace library as indicated in FIG. 6.

At this point the properties of the trace library should be evident and the dynamics of its use in a multiprocessing environment can be illustrated by the diagram presented in FIG. 2. As indicated in FIG. 2, several processes in a multiprocessing environment, only two of which are shown as process 210 and 220, can be simultaneously providing outputs from their trace library to trace data files that are uniquely associated with each of the processes. The overall action of the debugging process for the individual processor, however, is controlled by the global trace table which is present in the processor at a point in its memory that is available to all operating processes. As indicated hereinabove, the present embodiment includes a global trace table within the part of the memory that is allocated to the VMS operating system. The global sync word is also present in a portion of memory that is available to each of the processes. All of the processes in this multiprocessing environment can continue to execute without re-initialization even though they are taken in and out of the debugging process. The parameters of the global trace table are simply changed in order to exclude or include specific processes or to include or exclude subsystems and functions within those processes. Finally, the intensity of the debugging and therefore the time consumed on the processor by the debugging trace facility can be changed by modifying the trace level within the global trace table. The trace facility can be completely deactivated by excluding all active processes from the process list. In summary, the present invention provides a trace facility that can dynamically interact with continuously executing processes in a multiprocessing environment by permeating the code of these processes to varying degrees depending on the parameters specified in the global trace table.

It should be readily apparent to those skilled in the art that numerous changes can be made in the present embodiment without departing from the spirit and scope of the present invention. For example, the information provided as an output from the trace library may also be delivered to a terminal in addition to, or in place of, the trace data file. The global and local trace tables may also include other lists that permit further specification

of the kind of code that is to cause an output from the trace facility. Obviously, additional levels may also be included in order to change the degrees to which the tracing facility is permitted to permeate the processes operating in the multiprocessing environment.

What is claimed is:

1. In a multiprocessing computer environment wherein several processes are simultaneously executing on a single processor, at least one process having calls to a trace library to provide an output when the arguments accompanying a call satisfy a predetermined criteria with respect to parameters within a table that is local to said at least one process, a computer method of controlling the execution of the trace statements within said at least one process comprising the steps of comparing a local word within the process with a global word available to all processes, and replacing the parameters in the local table with their corresponding values from a global table that is available to all processes in response to an indication that a difference exists between said local word and said global word.

2. A computer method as defined in claim 1 wherein the global table includes a list of processes that are to be acted upon by said trace facility and each process includes a process flag, the method further including the steps of comparing the calling process name to the processes present in the process list, and setting the process flag to a predefined condition if the process name is present in the process list.

3. A computer method as defined in claim 2 wherein each local and global table further includes a trace level parameter, and each call to the trace library further includes a level-indicating argument, the method further including the steps of comparing the level-indicating argument in the calling code to the trace level parameter present in the local table, and returning without producing an output from the trace library when the comparison to the trace level parameter indicates a predetermined relationship.

4. A method for operating a tracing facility within a multiprocessing computer environment wherein each of several processes has a trace library that includes a local sync word and a local trace table and the processor environment includes a global sync word and a global trace table that are available to each of the processes operating in the multiprocessing environment, both local trace table and global trace table including a list of parameters that indicate which encoded trace statements are to be acted upon, said method comprising the steps of comparing the local sync word with the global sync word, and replacing the parameters of the local trace table with their equivalent values from the parameters in the global trace table in response to an indication by the comparison of the local sync word and global sync word that a predetermined difference exists between the two sync words.

5. A computer method as defined in claim 4 wherein each process includes a process flag and the method further includes the additional steps of comparing the process name with the parameters listed in said local trace table to determine whether the process is included within those parameters, and setting the process flag to a predetermined state if the comparison determines that the process name is present within those parameters.

6. A computer method as defined in claim 4 wherein each local trace table and the global trace table includes a parameter to indicate a trace level and each calling code to the trace facility includes a level-indicating